

分散演算を用いたパイプラインLMS適応デジタルフィルタの高性能アーキテクチャ

High-performance Architecture of the Pipelined LMS Adaptive Filter using Distributed Arithmetic

橋内慎次郎[†], ○内田勝也[†], 佐藤 慎悟[†], 高橋 強^{††}, 恒川 佳隆[†]

Shinjiro KITSUNAI[†], ○Katsuya UCHIDA[†], Shingo SATO[†], Kyo TAKAHASHI^{††},
Yoshitaka TSUNEKAWA[†]

[†]岩手大学, ^{††}岩手県工業技術センター

[†]Iwate University, ^{††}Iwate Industrial Research Institute

キーワード: 分散演算(distributed arithmetic), LMSアルゴリズム(LMS algorithm), 同時更新(simultaneous update), パイプライン・アーキテクチャ(pipelined architecture), 高サンプリング・レート(high-sampling rate)

連絡先: 〒020-8551 盛岡市上田4-3-5 岩手大学工学部

橋内慎次郎, Tel.:(019)621-6468, Fax.:(019)621-6468, E-mail: t3308006@iwate-u.ac.jp,

1. はじめに

適応フィルタは、エコーキャンセラ、ノイズコントローラ、適応等化器、振動制御などさまざまな分野で用いられ、ますます応用範囲を広げている。適応フィルタを実現するには、高速なサンプリングレート、良好な収束特性、短い出力滞在時間、小規模ハードウェア、低消費電力などさまざまな性能が要求されるが、これらを同時に満たすことは困難であり、高性能なアルゴリズムや効果的なアーキテクチャが望まれている。

報告者らは、ベクトルの内積演算を効率良く実行する分散演算(Distributed Arithmetic, DA)をLMS適応フィルタ¹⁾に適用して分散演算形LMS適応アルゴリズム(DA-LMS)と効果的なアーキテクチャを提案した^{2) 3)} この手法では、フィルタリングにおける係数ベクトルと入力信号ベクトルの内積演算に分散演算を適用することにより、比較的規模

の乗算器を用いず、加算器とレジスタで適応フィルタを構成することが可能である。これより、分散演算形LMS適応フィルタ(LMS Adaptive Filter using Distributed Arithmetic, DA-ADF)は、小規模ハードウェア、低消費電力であり、しかも良好な収束特性を示す高性能な適応フィルタである。

しかし、部分積のシフト加算を語長回数だけ繰り返すDA-ADFは、これまでの乗算器を用いたパイプライン適応フィルタ^{4) 5) 6)}と比較するとサンプリングレートが低く、適用範囲が限定される。

本報告では、DA-LMSの高速アルゴリズムと効果的なVLSIアーキテクチャを提案する。提案するアルゴリズムは、適応フィルタの基本処理である出力計算と更新動作を同時並列に実行することにより、処理時間の短縮を可能にする。収束速度を計算機シミュレーションにより評価し、更新に1時刻前の誤差信号を用いているにも関わらず、遅延

時間による収束速度の劣化は非常に小さいことを示す。さらに、提案するアルゴリズムを効果的に実現するVLSIアーキテクチャを示す。

2. 分散演算形LMS適応フィルタ

2.1 分散演算の適用

従来、分散演算は定係数ベクトルの内積演算を効率よく実行する手法として知られていたが、係数が時間変化する適応フィルタにおいても効果的である。以下、フィルタのタップ数を N 、入力信号語長を B として議論を進める。

図1にDA-ADFの基本構成を示す。分散演算は、適応フィルタの出力計算における N 次係数ベクトル $\mathbf{W}(k)$ と N 次入力信号ベクトル $\mathbf{S}(k)$ の内積演算

$$y(k) = \mathbf{W}^T(k) \mathbf{S}(k)$$

$$\mathbf{W}(k) = [w_0(k), \dots, w_{N-1}(k)]^T$$

$$\mathbf{S}(k) = [s(k), s(k-1), \dots, s(k-N+1)]^T (1)$$

に対して適用される。入力信号はLSBから入力され、この1ビットと $N-1$ bitのシフトレジスタ出力がRAMのアドレス信号(アドレスベクトル)となる。このアドレスベクトルに対して N 次係数ベクトルの部分積が定義され、RAM(Random Access Memory)に格納される。部分積の数は、 N 次アドレスベクトルに対して 2^N 個が存在し、これらの部分積を要素とする集合を全適応関数空間(Whole Adaptive Function Space, WAFS)と呼ぶ。出力計算は、決定された B 個のアドレスベクトルに対する部分積をRAMから順に選択して読み出し、シフト加算を実行する。更新動作は、求めた誤差信号 $e(k)$ を用いて出力計算に使用した部分積を順に更新する。ある時刻 k において選択される B 個の部分積の集合を適応関数空間(Adaptive Function Space, AFS)と呼ぶ。

DA-LMSアルゴリズムを以下に示す。入力信号

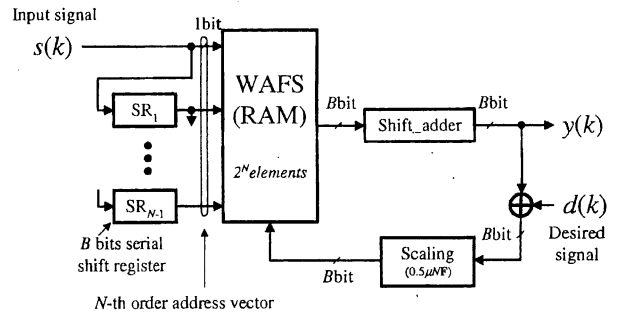


Fig. 1 Basic structure of DA-ADF

は2の補数形式を用いて次式のように表される。

$$s(k) = -2^0 \cdot b_0(k) + 2^{-1} \cdot b_1(k) + \dots + 2^{-B+1} \cdot b_{B-1}(k)$$

ここで、 $b_i(k)$, $i = 0, \dots, B-1$ は $x(k)$ の i 番目のビット値を表す。式(1)の N 次入力信号ベクトルは

$$\mathbf{S}(k) = \mathbf{A}^T(k) \mathbf{F}$$

と表される。なお、 T はマトリクスの転置を表す。これをビットパターンに分解した $N \times B$ 次アドレスマトリクスは

$$\mathbf{A}(k) = \begin{bmatrix} b_0(k) & \dots & b_0(k-N+1) \\ b_1(k) & \dots & b_1(k-N+1) \\ \vdots & \ddots & \vdots \\ b_{B-1}(k) & \dots & b_{B-1}(k-N+1) \end{bmatrix}^T, \quad (2)$$

スケーリングベクトル \mathbf{F} は

$$\mathbf{F} = [-2^0, 2^{-1}, \dots, 2^{-(B-1)}]^T$$

である。式(2)の列ベクトルである B 個のアドレスベクトルで指定される適応関数空間 $\mathbf{P}(k)$

$$\begin{aligned} \mathbf{P}(k) &= [p_0(k), p_1(k), \dots, p_{B-1}(k)]^T \\ &= \mathbf{A}^T(k) \mathbf{W}(k) \end{aligned}$$

を用いて、フィルタ出力 $y(k)$ は以下のように表される。

$$y(k) = \mathbf{F}^T \mathbf{P}(k)$$

また、AFSの更新式は

$$\mathbf{P}(k+1) = \mathbf{P}(k) + 0.5\mu N e(k) \mathbf{F}$$

である。ここで、誤差信号は

$$e(k) = d(k) - y(k)$$

であり、 $d(k)$ は所望信号を表す。

2.2 マルチメモリブロック構成

DA-ADFは、高次においてWAFSを実現するRAMの容量が膨大となりハードウェア規模と消費電力の点で不利である。また、膨大な部分積数を有するWAFSを更新することは、収束状態に達するまでに多くの繰り返しを必要とし収束速度が緩慢になる。この問題を解決するために、WAFSを N 方向に M 分割するマルチメモリブロック構造(Multi-memory block structure)を適用する。DA-ADFにマルチメモリブロック構造を適用した分散演算形LMS適応フィルタをMDA-ADFと呼ぶ。図2にMDA-ADFの基本構成、図3にタイミングチャートを示す。分割数 M 、アドレスベクトルの次数 $R (= N/M)$ のMDA-ADFは、要素数 2^R 個のWAFSを M 個有し、WAFSの総容量は $M \times 2^R$ wordである。また、分割数 M が大きく R が小さいほどWAFSの要素数が減少するため、収束速度が向上する。

アルゴリズムを以下に示す。 M 分割された m 番目のアドレスマトリクス $\mathbf{A}_m(k)$ は

$$\mathbf{A}_m(k) = \begin{bmatrix} b_{m0}(k) & \cdots & b_{m0}(k-R+1) \\ b_{m1}(k) & \cdots & b_{m1}(k-R+1) \\ \vdots & \ddots & \vdots \\ b_{m(B-1)}(k) & \cdots & b_{m(B-1)}(k-R+1) \end{bmatrix}^T$$

と表される。この $\mathbf{A}_m(k)$ に対する AFS である $\mathbf{P}_m(k)$ は次式で表される。

$$\mathbf{P}_m(k) = [p_{m0}(k), p_{m1}, \cdots, p_{m(B-1)}(k)]^T$$

$$m = 0, 1, \cdots, M-1$$

フィルタ出力は分割されたAFSを用いた個別の出

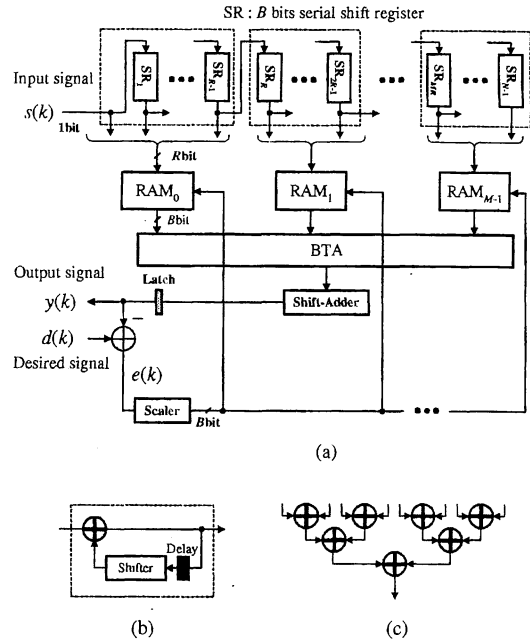


Fig. 2 Basic structure of MDA-ADF

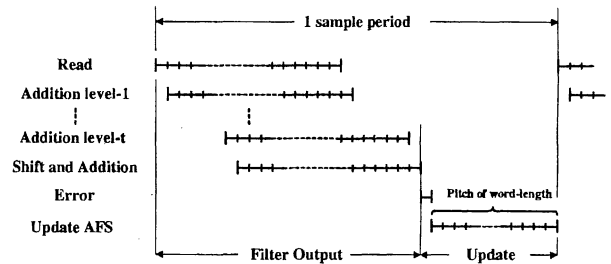


Fig. 3 Timing chart of the MDA-ADF

力の総和により得られ、次式で求められる。

$$y(k) = \sum_{m=0}^{M-1} \mathbf{F}^T \mathbf{P}_m(k) \quad (3)$$

また、分割された m 番目のAFSに対する更新式は次のように表される。

$$\mathbf{P}_m(k+1) = \mathbf{P}_m(k) + 0.5\mu Re(k) \mathbf{F}$$

$$\mathbf{P}_m(k) = [p_{m0}(k), p_{m1}, \cdots, p_{m(B-1)}(k)]^T$$

$$m = 0, 1, \cdots, M-1$$

3. 同時更新アルゴリズム

MDA-ADFの高速アルゴリズムを検討する。MDA-ADFは出力計算、誤差計算、更新動作の各

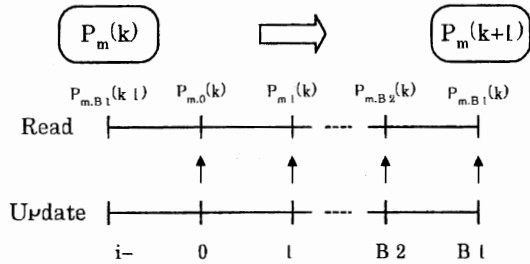


Fig. 4 Read and Update procedure

動作をシーケンシャルに実行するが、1時刻前の誤差信号を用いて出力計算と更新動作を並列に実行（パイプライン化）することを考える。過去の誤差信号を用いて係数を更新する手法は、ディレードアップデートと呼ばれ、収束速度の劣化を許容して適応フィルタをパイプライン化する手法として知られている^{7) 8) 9)}。この手法では、過去の誤差を用いて係数を更新した後出力計算を実行する。しかし、提案法におけるディレードアップデートでは、分散演算の特徴である出力計算における B 回のAFSの読み出しと更新動作における B 回の更新動作を同時に実行する。この際、出力計算は重みの小さいアドレスベクトル、更新動作は重みの大きいアドレスベクトルから更新動作を実行する。この提案するアルゴリズムを同時更新アルゴリズムと呼び、WAFSの読み出し・更新の様子を図4に示す。なお、このアルゴリズムを用いた適応フィルタを同時更新形MDA-ADF(MDA-ADF using Simultaneous update, SMDA-ADF)と呼ぶことにする。提案する同時更新アルゴリズムを以下に示す。フィルタ出力 $y(k)$ と分割された適応関数空間の出力 $y_m(k)$ は次式で表される。

$$y(k) = \sum_{m=0}^{M-1} y_m(k),$$

$$y_m(k) = \sum_{i=0}^{B-1} \mathbf{F}_{B-1-i} \mathbf{P}_{m,B-1-i}(k).$$

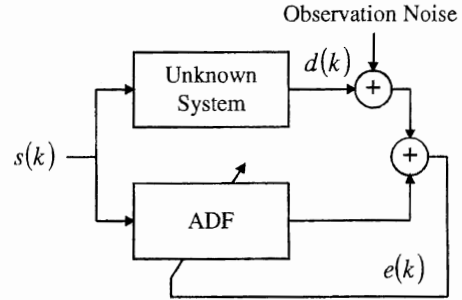


Fig. 5 Simulation model

ここで、 \mathbf{F}_i は以下に示す i 番目の要素以外は0のスケールリングベクトルである。

$$\mathbf{F}_0 = [-2^0, 0, 0, \dots, 0, 0]^T,$$

$$\mathbf{F}_1 = [0, 2^{-1}, 0, \dots, 0, 0]^T,$$

$$\vdots$$

$$\mathbf{F}_{B-1} = [0, 0, \dots, 0, 2^{-B+1}]^T.$$

ここで、 $\mathbf{P}_{m,i}(k)$ は M 分割された m 番目のAFSであり、添え字の i は時刻 k における $i(i=0, 1, \dots, B-1)$ 番目に更新されたことを表す。ここで、

$$\mathbf{P}_m(k) \equiv \mathbf{P}_{m,B-1}(k-1)$$

である。更新式は次式で表される。

$$\mathbf{P}_{m,i}(k+1) = \mathbf{P}_{m,i}(k) + 0.5\mu Re(k-1)\mathbf{F}_i,$$

$$i = 0, 1, \dots, B-1. \quad (4)$$

なお、誤差信号 $e(k)$ は

$$e(k) = d(k) - y(k) \quad (5)$$

である。

4. 収束特性

システム同定問題に対する従来のMDA-ADFとSMDA-ADFの収束特性を比較する。シミュレーションモデルを図5に示す。ここで、未知システムは32タップ低域通過FIRフィルタ、入力信号は平均0.0、分散1.0の白色ガウス雑音、観測信号は平均

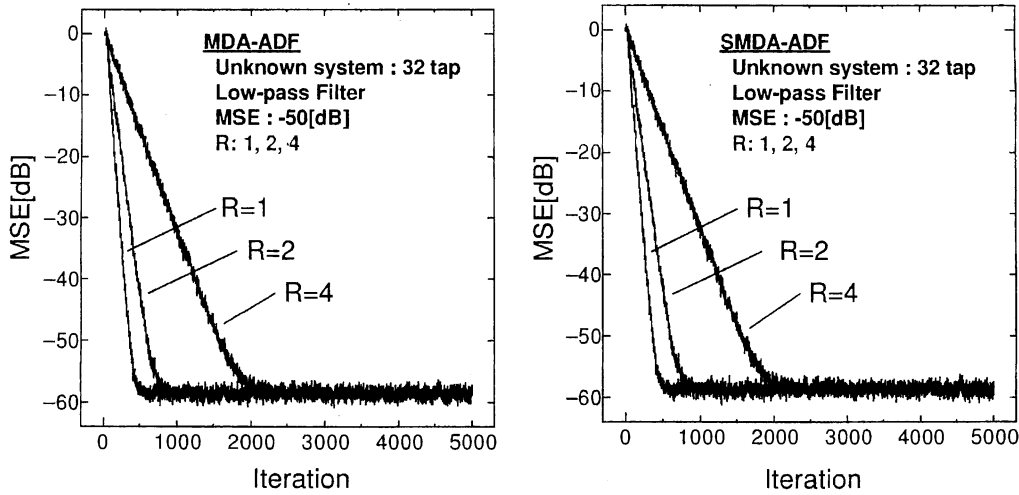


Fig. 6 Comparison of the convergence characteristics between MDA and SMDA algorithm (a) MDA-ADF (b) SMDA-ADF

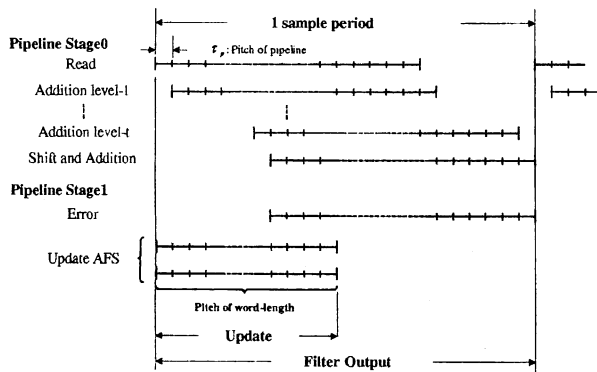


Fig. 8 Timing chart of the proposed Architecture

0.0, 分散 10^{-6} の入力信号とは無相関の白色ガウス雑音を加えた。また, ステップサイズパラメータはMSEが $-58.5[\text{dB}]$ を示す値を選択し, WAFSの分割数は $M=32, 16, 8 (R=1, 2, 4)$ とした。収束特性を図6に示す。MDA, SMDAともに, 小さい R に対して高速な収束速度を示していることがわかる。提案するSMDA-ADFは従来のMDA-ADFと同等の良好な収束特性を示すことがわかる。

5. アーキテクチャ

提案するVLSIアーキテクチャを図7, タイミングチャートを図8に示す。入力信号レジスタは

$(N-1) \times B$ 個のレジスタを内部に有し, 適応関数空間の要素を指定するアドレスマトリクスを M 個に分割された適応関数空間モジュールに供給する。 M 個の適応関数空間モジュール (AFSM) は, 2^R 個のラッチ, セレクタ, コントローラ, 加算器から構成され, フィルタ出力計算に用いる関数空間の要素の読み出しと, 更新のための読み出し書き込みを行う。コントローラは, アドレスマトリクスから1つのアドレスベクトルを選択し, 選択信号をセレクタに供給する。なお, 出力計算は最小重みを持つアドレスベクトルから順次実行する。適応関数空間は, 従来法ではRAMを使用していたが, 提案法では図8の様に読み出しと書き込みを同時に実行することが可能な構成を用いている。 M 個の適応関数空間からの出力はバイナリツリーアダ (BTA) により加算され, シフト加算器を用いて出力を計算する。また, 誤差の計算は出力計算と並列に実行される。この際, シフトレジスタには初期値として所望信号 $d(k)$ と符号反転の際に加える値'1'をLSBに与えている。これにより, 提案法ではフィルタ出力と誤差信号を同時に求めることが可能になる。以上で, フィルタ出力と誤差信号を求めるパイプラインステージは動作を完了す

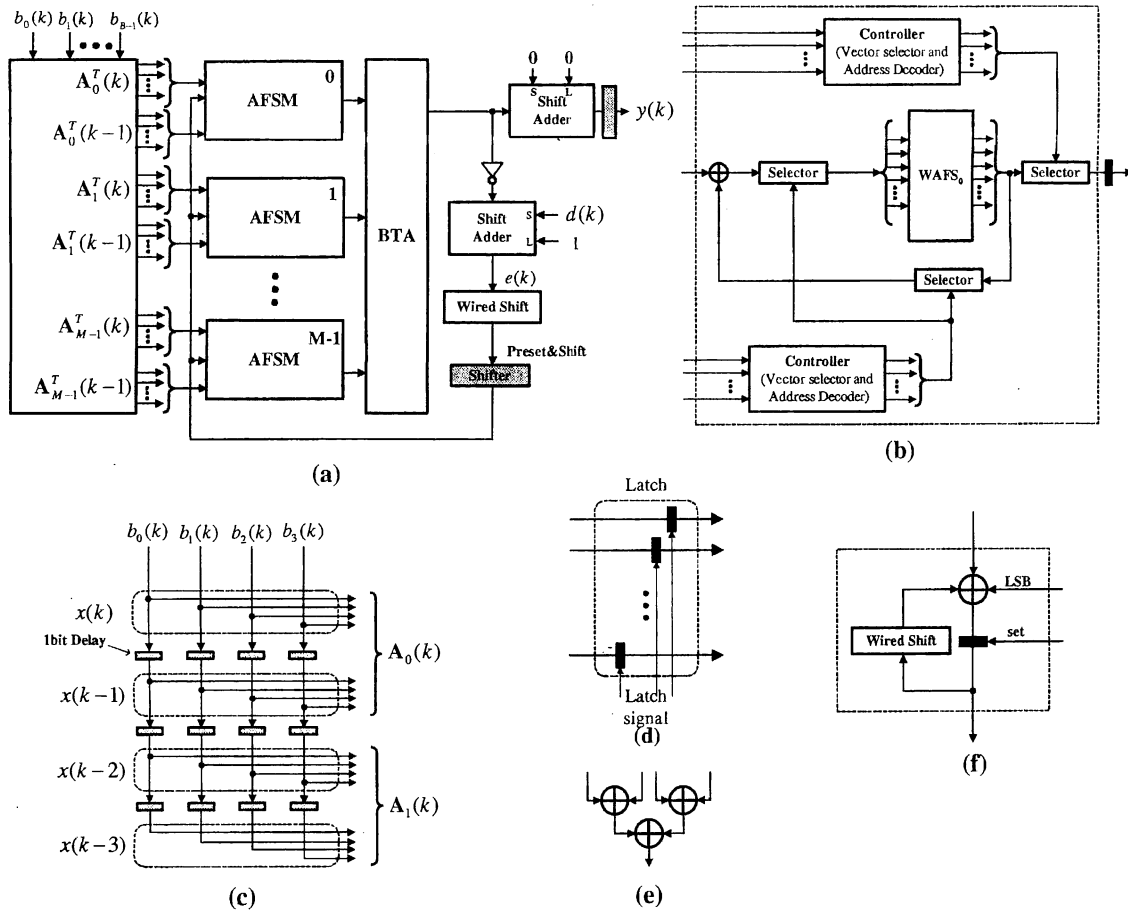


Fig. 7 Proposed VLSI architecture (a) Block diagram (b) Adaptive function space module (AFSM) (c) Example of the input register for $N=4, R=2, B=4$ (d) Structure of the Adaptive function space (e) Example of the binary-tree adder (BTA) for 4-input (f) Structure of the shift-adder

る。次いで、適応関数空間は出力計算のための読み出しと並列にシフトされた誤差信号を用いて更新される。更新動作は、最大重みを持つアドレスベクトルから順次実行される。

6. まとめ

本報告では、従来の分散演算形LMS適応フィルタの高速アルゴリズムと効果的アーキテクチャを提案した。収束特性は従来のMDA-ADFと同等の良好な収束速度を示すことを計算機シミュレーションにより検証した。

今後の課題は、収束速度の詳細な評価を進めるとともに、収束条件の理論的解析を行うこと、提案したVLSIアーキテクチャのVLSI評価を行うこ

とである。

参考文献

- 1) B. Widrow and M. E. Hoff, "Adaptive Switching Circuit," IRE EWSCON Conv. Rec., pp96-104, 1960.
- 2) 恒川, 高橋, 豊田, 三浦, "分散演算によるマルチプライヤレスLMS適応フィルタの高性能アーキテクチャ," 信学論(A), vol.J-82-A, no.10, pp.1518-1528, Oct. 1999.
- 3) 高橋, 恒川, 豊田, 三浦, "ハーフメモリアルゴリズムに基づく分散演算形LMS適応フィルタの高性能アーキテクチャ," 信学論(A), Vol. J-84-A No.6 pp.777-787, June 2001.
- 4) K.J. Raghunath and K.K. Parhi, "A 100 MHz pipe-lined RLS adaptive filter," Proc. IEEE ICASSP'95, Detroit, Michigan, pp.3187-3190, May 1995.
- 5) 松原, 西川, 貴家, "Delayed LMS アルゴリズムに基づくパイプライン適応フィルタ," 信学論(A), vol.J79-A, no.5, pp.1050-1057, May 1996.

- 6) A. Harada, K. Nishikawa and H. Kiya, "Pipelined Architecture of the LMS Adaptive Digital Filter with the Minimum Output Latency," IEICE Trans. Fundamentals, pp.1578-1585, Vol.E81-A No.8 1998
- 7) R. Haimi-Cohen, H. Herzberg, and Y. Be'ery, "Delayed adaptive LMS filtering: Current results," Proc. IEEE ICASSP '90, pp.1273-1276, April 1990.
- 8) G. Long, F. Ling, and J.G. Proakis, "The LMS algorithm with delayed coefficient adaptation," IEEE Trans. Acoust. Speech Signal Process., vol.37, no.9, pp.1397-1405, Sept. 1989.
- 9) G. Long, F. Ling, and J.G. Proakis "Corrections to "The LMS algorithm with delayed coefficient adaptation", " IEEE Trans. Acoust. Speech Signal Process., vol.40, no.1, pp.230-232, Jan. 1992.